

IP 组播故障排除指南

目录

[简介](#)

[先决条件](#)

[要求](#)

[使用的组件](#)

[背景信息](#)

[由于 RPF 故障，路由器未向主机转发多播数据包](#)

[诊断问题](#)

[可能的修正](#)

[由于源端 TTL 设置，路由器未向主机转发多播数据包](#)

[诊断问题](#)

[可能的修正](#)

[由于路由器的 TTL 阈值，路由器未转发多播数据包](#)

[诊断问题](#)

[可能的修正](#)

[多条相等成本路径导致不需要的 RPF 行为](#)

[诊断问题](#)

[可能的修正](#)

[为什么没有在所有可用的等价路径之间进行 IP 多播负载均衡？](#)

[可能的修正](#)

[为什么在路由器上收到 IP 多播“INVALID RP JOIN”错误消息？](#)

[诊断问题 - 第 1 部分](#)

[可能的修正](#)

[诊断问题 - 第 2 部分](#)

[可能的修正](#)

[CGMP 不能防止多播数据包泛洪](#)

[诊断问题](#)

[观察](#)

[可能的修正](#)

[由于源端/接收端的放置问题，CGMP 不能防止多播数据包泛洪](#)

[诊断问题](#)

[可能的修正](#)

[CGMP 不能防止某些组地址的多播数据包泛洪](#)

[可能的修正](#)

[收到重复的多播数据包流](#)

[原因 1](#)

[可能的修复方法 1](#)

[原因 2](#)

[可能的修复方法 2](#)

[原因 3](#)

[可能的修复方法 3](#)

[多播数据包为什么被丢弃？](#)

[原因 1](#)

[可能的修复方法 1](#)

[原因 2](#)

[可能的修复方法 2](#)

[相关信息](#)

简介

本文介绍 IP 多播的常见问题和解决方案。

先决条件

要求

本文档没有任何特定的要求。

使用的组件

本文档不限于特定的软件和硬件版本。

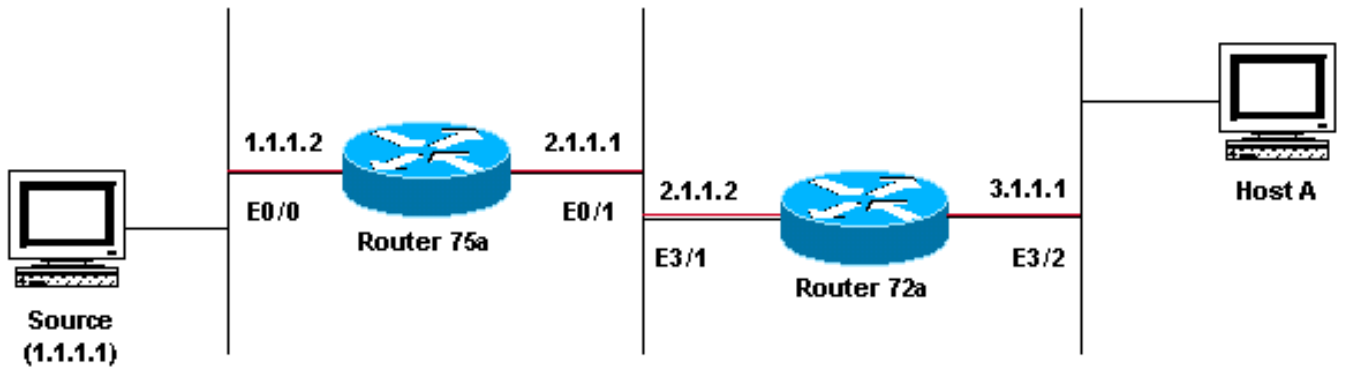
背景信息

当您排除故障组播路由时，最关心是源地址。组播有反向路径转发(RPF)检查的概念。当组播信息包在接口时到达，保证RPF的过程检查此流入接口是流出接口由单播路由使用为了到达组播信息包的来源。此 RPF 检查进程将防止出现环路。除非数据包的来源通过RPF检查，组播路由不转发数据包。数据包通过此 RPF 检查后，多播路由将仅根据目标地址转发数据包。

类似单播路由，组播路由有几份可用的协议，例如独立于协议的组播密集模式(PIM-DM)，PIM稀疏模式(PIM-SM)，距离矢量组播路由协议(DVMRP)、组播边界网关协议(MBGP)和多播源发现协议(MSDP)。在本文的案例研究通过排除故障多种问题的进程走您。您将看到哪些命令用于为了迅速精确定位问题和学习如何解决它。这里列出的案例研究适用于各种协议，除非特别说明。

由于 RPF 故障，路由器未向主机转发多播数据包

此部分提供一解决方案给IP组播RPF故障的常见问题。我们以下面的网络图为例。



Sending to 224.1.1.1

在此图，组播信息包进入路由器75a E0/0从IP地址是1.1.1.1并且发送分组224.1.1.1的服务器的。这称为 (S,G) 或 (1.1.1.1, 224.1.1.1)。

诊断问题

直接连接到路由器 75a 的主机将接收该多播馈送，但直接连接到路由器 72a 的主机不会进行接收。首先，请输入 **show ip mroute 224.1.1.1** 命令为了用路由器75a发现怎么回事。该命令将检查组地址 224.1.1.1 的多播路由 (mroute)：

```
75a#show ip mroute 224.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:01:23/00:02:59, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse-Dense, 00:01:23/00:00:00

(1.1.1.1, 224.1.1.1), 00:01:23/00:03:00, flags: TA
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse-Dense, 00:01:23/00:00:00
```

因为路由器运行PIM密集模式(您知道它是密集模式由于D标志)，请忽略*，G条目和重点在S，G条目。该条目表明多播数据包来自地址为 1.1.1.1 的服务器并发送到多播组 224.1.1.1。数据包进入 Ethernet0/0接口和转发Ethernet0/1接口。这是一个完美的方案。

输入 **show ip pim neighbor** 命令为了发现路由器72a是否显示上游路由器(75a)作为PIM邻居：

```
ip22-72a#show ip pim neighbor
PIM Neighbor Table
Neighbor Address  Interface      Uptime    Expires    Ver  Mode
2.1.1.1          Ethernet3/1    2d00h     00:01:15  v2
```

从 **show ip pim neighbor** 命令输出，PIM邻居看起来好。

输入 **show ip mroute** 命令为了发现路由器72a是否有好mroute：

```
ip22-72a#show ip mroute 224.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, B - Bidir Group, s - SSM Group, C - Connected,
```

L - Local, P - Pruned, R - RP-bit set, F - Register flag,
T - SPT-bit set, J - Join SPT, M - MSDP created entry,
X - Proxy Join Timer Running, A - Candidate for MSDP Advertisement,
U - URD, I - Received Source Specific Host Report, Z - Multicast Tunnel
Y - Joined MDT-data group, y - Sending to MDT-data group
Outgoing interface flags: H - Hardware switched, A - Assert winner
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

```
(*, 224.1.1.1), 00:10:42/stopped, RP 0.0.0.0, flags: DC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet3/1, Forward/Dense, 00:10:42/00:00:00
    Ethernet3/2, Forward/Dense, 00:10:42/00:00:00
```

```
(1.1.1.1, 224.1.1.1), 00:01:10/00:02:48, flags:
  Incoming interface: Ethernet2/0, RPF nbr 0.0.0.0 Outgoing interface list: Ethernet3/1,
  Forward/Dense, 00:01:10/00:00:00 Ethernet3/2, Forward/Dense, 00:00:16/00:00:00 ip22-72a#
```

通过 **show ip mroute 224.1.1.1** 命令可以发现，传入接口为 Ethernet2/0 而不是预期的 Etheret3/1。

输入**show ip mroute count**命令的**224.1.1.1**为了发现此组的任何组播数据流是否到达到路由器 72a，并且什么其次发生：

```
ip22-72a#show ip mroute 224.1.1.1 count
IP Multicast Statistics
3 routes using 2032 bytes of memory
2 groups, 0.50 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg
Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null,
rate-limit etc)
```

```
Group: 224.1.1.1, Source count: 1, Packets forwarded:      0, Packets received: 471
  Source:      1.1.1.1/32, Forwarding: 0/0/0/0, Other: 471/471/0 ip22-72a#
```

从 Other 计数中可以看到流量由于 RPF 故障而被丢弃：total 471 drops, due to RPF failure - 471...

输入**show ip rpf <source>**命令为了发现是否有RPF错误：

```
ip22-72a#show ip rpf 1.1.1.1
RPF information for ? (1.1.1.1)
  RPF interface: Ethernet2/0
  RPF neighbor: ? (0.0.0.0)
  RPF route/mask: 1.1.1.1/32
  RPF type: unicast (static)
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
ip22-72a#
```

Cisco IOS® 以这种方式计算 RPF 接口。RPF信息可能的来源是单播路由表、MBGP路由表、DVMRP路由表和静态Mroute表。当您计算RPF接口时，信息源RPF计算根据主要的管理距离用于为了正确地确定。具体规则如下：

- 在所有以前的 RPF 数据源中搜索源 IP 地址的匹配项。当您使用共享结构树时，RP地址使用而不是源地址。
- 如果找到多个匹配路由，则使用管理距离最短的路由。
- 如果管理距离是相等的，则使用此优先级顺序：静态多播路由DVMRP 路由MBGP 路由单播路由
- 如果某个路由在同一个路由表中有多个条目，则使用最长的匹配路由。

show ip rpf 1.1.1.1 命令输出表明 RPF 接口为 E2/0，但传入接口应该是 E3/1。

输入 `show ip route 1.1.1.1` 命令为了发现RPF接口为什么是与什么不同预计了。

```
ip22-72a#show ip route 1.1.1.1
Routing entry for 1.1.1.1/32
Known via "static", distance 1, metric 0 (connected)
Routing Descriptor Blocks:
* directly connected, via Ethernet2/0
Route metric is 0, traffic share count is 1
```

从此 `show ip route 1.1.1.1` 命令的输出中可以看到有一个静态 /32 路由，它导致选择了错误的接口作为 RPF 接口。

输入一些进一步调试指令：

```
ip22-72a#debug ip mpacket 224.1.1.1
*Jan 14 09:45:32.972: IP: s=1.1.1.1 (Ethernet3/1)
d=224.1.1.1 len 60, not RPF interface
*Jan 14 09:45:33.020: IP: s=1.1.1.1 (Ethernet3/1)
d=224.1.1.1 len 60, not RPF interface
*Jan 14 09:45:33.072: IP: s=1.1.1.1 (Ethernet3/1)
d=224.1.1.1 len 60, not RPF interface
*Jan 14 09:45:33.120: IP: s=1.1.1.1 (Ethernet3/1)
d=224.1.1.1 len 60, not RPF interface
```

数据包在E3/1进来，正确。然而，他们丢弃，因为那不是单播路由表使用RPF检查的接口。

注意：调试数据包具有一定危险。数据包调试触发处理组播信息包的交换，强化中央处理。此外，数据包调试还会产生大量输出，由于向控制台端口输出过慢，从而可能导致将路由器完全挂起。在您调试数据包前，必须保重特别注意为了禁用操作日志输出到控制台和启用日志到存储器缓冲区。为此，需配置 `no logging console` 和 `logging buffered debugging`。可以使用 `show logging` 命令查看调试结果。

可能的修正

您能或者更改单播路由表为了满足此要求或您能添加一静态mroute驱逐组播到RPF特定接口，不管什么单播路由表陈述。添加静态多播路由：

```
ip22-72a(config)#ip mroute 1.1.1.1 255.255.255.255 2.1.1.1
```

此静态mroute声明那达到RPF的地址1.1.1.1，使用2.1.1.1作为接口E3/1的下一跳。

```
ip22-72a#show ip rpf 1.1.1.1
RPF information for ? (1.1.1.1)
RPF interface: Ethernet3/1
RPF neighbor: ? (2.1.1.1)
RPF route/mask: 1.1.1.1/32
RPF type: static mroute
RPF recursion count: 0
Doing distance-preferred lookups across tables
```

`show ip mroute`和`debug ip mpacket`输出看似好，被发送的数据包数量在`show ip mroute count`的增加和HostA接收数据包。

```
ip22-72a#show ip mroute 224.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
M - MSDP created entry, X - Proxy Join Timer Running
```

```
A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:01:15/00:02:59, RP 0.0.0.0, flags: DJC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet3/1, Forward/Sparse-Dense, 00:01:15/00:00:00
    Ethernet3/2, Forward/Sparse-Dense, 00:00:58/00:00:00
```

```
(1.1.1.1, 224.1.1.1), 00:00:48/00:02:59, flags: CTA
  Incoming interface: Ethernet3/1, RPF nbr 2.1.1.1, Mroute
  Outgoing interface list:
    Ethernet3/2, Forward/Sparse-Dense, 00:00:48/00:00:00
```

```
ip22-72a#show ip mroute 224.1.1.1 count
IP Multicast Statistics
3 routes using 2378 bytes of memory
2 groups, 0.50 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 224.1.1.1, Source count: 1, Packets forwarded: 1019, Packets received: 1019
  Source: 1.1.1.1/32, Forwarding: 1019/1/100/0, Other: 1019/0/0
```

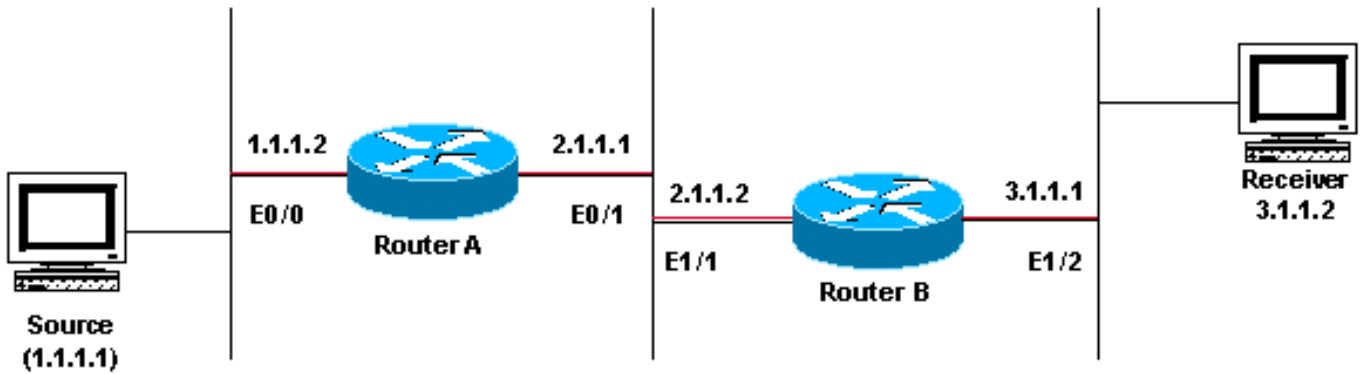
```
ip22-72a#show ip mroute 224.1.1.1 count
IP Multicast Statistics
3 routes using 2378 bytes of memory
2 groups, 0.50 average sources per group
Forwarding Counts: Pkt Count/Pkts per second/Avg Pkt Size/Kilobits per second
Other counts: Total/RPF failed/Other drops(OIF-null, rate-limit etc)

Group: 224.1.1.1, Source count: 1, Packets forwarded: 1026, Packets received: 1026
  Source: 1.1.1.1/32, Forwarding: 1026/1/100/0, Other: 1026/0/0
ip22-72a#
```

```
ip22-72a#debug ip mpacket 224.1.1.1
*Jan 14 10:18:29.951: IP: s=1.1.1.1 (Ethernet3/1)
d=224.1.1.1 (Ethernet3/2) len 60, mforward
*Jan 14 10:18:29.999: IP: s=1.1.1.1 (Ethernet3/1)
d=224.1.1.1 (Ethernet3/2) len 60, mforward
*Jan 14 10:18:30.051: IP: s=1.1.1.1 (Ethernet3/1)
d=224.1.1.1 (Ethernet3/2) len 60, mforward
```

由于源端 TTL 设置，路由器未向主机转发多播数据包

此部分提供一解决方案给的IP组播数据包常见问题没有转发，因为存活时间(TTL)值消耗到零。由于多播应用的情况较多，因此这个问题很常见。多播应用主要设计用于 LAN，因此会将 TTL 设置为一个较低的值，甚至为 1。我们以下面的网络图为例。



Sending to 224.1.1.1

诊断问题

在上一个图，路由器A不转发数据包从来源到路由器B的直接连接的接收器。查看输出show ip mroute命令在路由器A为了发现组播路由状态：

```
ROUTERA#show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 00:00:01/00:00:00, RP 0.0.0.0, flags: DJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse-Dense, 00:00:01/00:00:00

(*, 224.1.1.1), 00:00:02/00:02:57, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse-Dense, 00:00:02/00:00:00
```

因为所有路由器参加此自动RP发现号组，您在输出中能忽略224.0.1.40。但是这里没有为 224.1.1.1 列出源。除了“*, 224.1.1.1”外，您还应当看到“1.1.1.1, 224.1.1.1”。

输入show ip rpf命令为了发现它是否是RPF问题：

```
ROUTERA#show ip rpf 1.1.1.1
RPF information for ? (1.1.1.1)
  RPF interface: Ethernet0/0
  RPF neighbor: ? (0.0.0.0) - directly connected
  RPF route/mask: 1.1.1.0/24
  RPF type: unicast (connected)
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
```

从输出，您看到它不是RPF问题。RPF 检查正确表明 E0/0 对应于源 IP 地址。

证实PIM是否在接口配置用show ip pim interface命令：

```
ROUTERA#show ip pim interface

Address          Interface          Version/Mode      Nbr   Query   DR
                  Count Intvl
1.1.1.2          Ethernet0/0        v2/Sparse-Dense  0     30     1.1.1.2
```

```
2.1.1.1      Ethernet0/1      v2/Sparse-Dense  2   30   2.1.1.2
```

输出一切正常，因此这不是问题所在。证实路由器是否认可任何有效数据流用**show ip mroute active**命令：

```
ROUTERA#show ip mroute active
Active IP Multicast Sources - sending >= 4 kbps
```

凭输出，路由器不认可任何有效数据流。

```
ROUTERA#debug ip mpacket
IP multicast packets debugging is on
```

或许接收方不发送任何互联网组管理协议(IGMP)报告(加入)组的224.1.1.1。您能用**show ip igmp group**命令检查它：

```
ROUTERB#show ip igmp group
IGMP Connected Group Membership
Group Address      Interface          Uptime    Expires    Last Reporter
224.0.1.40         Ethernet1/1        00:34:34 never      2.1.1.2
224.1.1.1          Ethernet1/2        00:30:02 00:02:45  3.1.1.2
```

224.1.1.1 已在 E1/2 加入，这是正确的。

PIM 密集模式是一个泛洪和修剪协议，因此没有加入，但是有嫁接。由于路由器 B 未从路由器 A 接收到泛洪，因此它并不知道向何处发送嫁接。

您可以使用嗅探器捕捉和 **show ip traffic** 命令进行检查以查看是否是 TTL 问题：

```
ROUTERA#show ip traffic
IP statistics:
  Rcvd: 248756 total, 1185 local destination
        0 format errors, 0 checksum errors, 63744 bad hop count
        0 unknown protocol, 0 not a gateway
        0 security failures, 0 bad options, 0 with options
```

输出显示63744坏跳数。每次键入此命令时，坏跳计数都将增加。这是明确的提示暗示发送方发送有TTL=1的数据包，路由器A减少到零。这导致坏跳计数字段不断增加。

可能的修正

为了解决问题，您需要增加TTL。这要在发送方的应用级别进行设置。有关详细信息，请参阅您的多播应用说明手册。

完成设置后，路由器 A 将如下所示：

```
ROUTERA#show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 01:16:32/00:00:00, RP 0.0.0.0, flags: DJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse-Dense, 01:16:32/00:00:00

(*, 224.1.1.1), 00:28:42/00:02:59, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
```



```

Outgoing interface list:
  Ethernet0/1, Forward/Sparse-Dense, 00:28:42/00:00:00
(1.1.1.1, 224.1.1.1), 00:19:24/00:02:59, flags: TA
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse-Dense, 00:19:24/00:00:00

```

这是您希望的输出。

在路由器 B 上，您会看到：

```

ROUTERB#show ip mroute
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.0.1.40), 01:23:57/00:00:00, RP 0.0.0.0, flags: DJCL
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/1, Forward/Sparse-Dense, 01:23:57/00:00:00

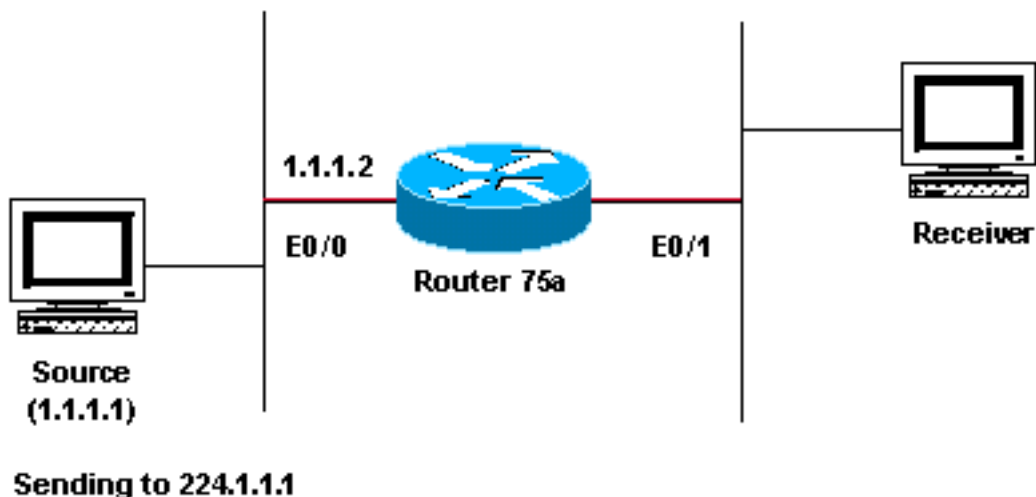
(*, 224.1.1.1), 01:19:26/00:02:59, RP 0.0.0.0, flags: DJC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/1, Forward/Sparse-Dense, 01:19:26/00:00:00
    Ethernet1/2, Forward/Sparse-Dense, 01:19:26/00:00:00

(1.1.1.1, 224.1.1.1), 00:17:46/00:02:59, flags: CTA
  Incoming interface: Ethernet1/1, RPF nbr 2.1.1.1
  Outgoing interface list:
    Ethernet1/2, Forward/Sparse-Dense, 00:17:46/00:00:00

```

由于路由器的 TTL 阈值，路由器未转发多播数据包

此部分提供一解决方案给TTL阈值设置的太低的常见问题，因此IP组播流量不到达接收方。我们以下的网络图为例。



诊断问题

在上一个图，接收方不从来源的接收组播数据包。也许有在来源和路由器75a之间的几路由器。首先查看路由器 75a，因为它直接连接到接收方。

```
ip22-75a#show ip mroute 224.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 00:32:05/00:02:59, RP 0.0.0.0, flags: DJC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse-Dense, 00:08:17/00:00:00

(1.1.1.1, 224.1.1.1), 00:01:02/00:01:57, flags: CTA
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/1, Forward/Sparse-Dense, 00:01:02/00:00:00
```

输出显示路由器75a转发数据包Ethernet0/1。为了是绝对肯定的路由器75a转发数据包，打开**调试**此来源和组播组的：

```
ip22-75a#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
ip22-75a(config)#access-list 101 permit udp host 1.1.1.1 host 224.1.1.1
ip22-75a(config)#end
ip22-75a#debug ip mpacket 101
IP multicast packets debugging is on for access list 101
ip22-75a#
*Jan 17 09:04:08.714: IP: s=1.1.1.1 (Ethernet0/0) d=224.1.1.1 len 60, threshold denied
*Jan 17 09:04:08.762: IP: s=1.1.1.1 (Ethernet0/0) d=224.1.1.1 len 60, threshold denied
*Jan 17 09:04:08.814: IP: s=1.1.1.1 (Ethernet0/0) d=224.1.1.1 len 60, threshold denied
```

调试消息表明路由器75a不转发组播信息包，因为TTL阈值达到了。查看路由器配置为了发现您是否能找到原因。下面的输出表明了问题所在：

```
interface Ethernet0/1
 ip address 2.1.1.1 255.255.255.0
 ip pim sparse-dense-mode
 ip multicast ttl-threshold 15
```

路由器的 TTL 阈值为 15，但这并不意味着任何大于 15 的 TTL 都不会予以发送。事实恰恰相反。应用程序用TTL 15传送。当它到达路由器 75a 时，多播数据包的 TTL 将小于 15。

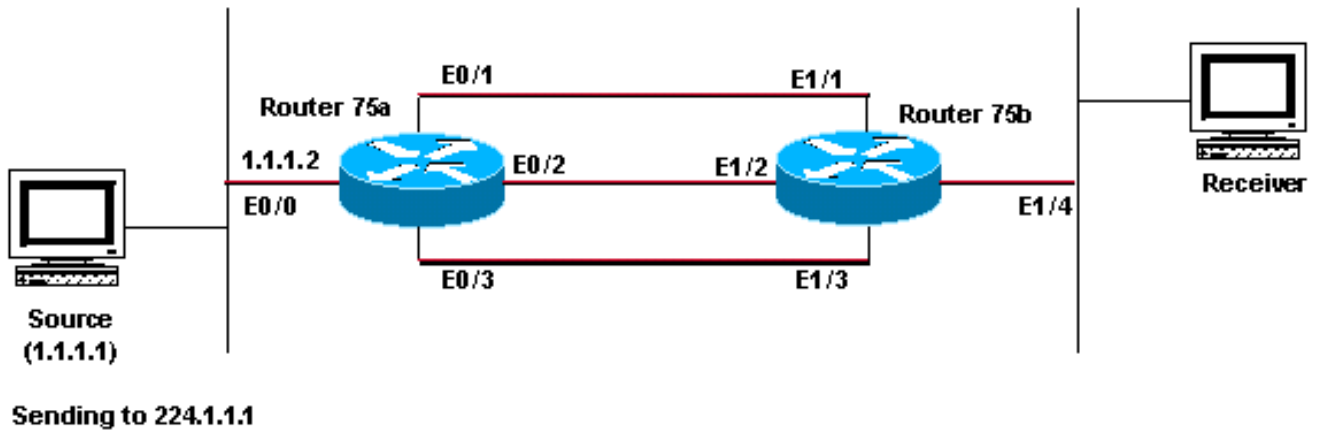
ip multicast ttl-threshold <value> 命令意味着 TTL 低于指定阈值（本例中为 15）的任何数据包都不会进行转发。此命令通常用于为了提供边界保持从漂移的内部组播数据流在内联网外面。

可能的修正

任一删除**IP组播TTL -门限<value>**命令用此命令**no**表示，恢复对默认TTL阈值为0，或者降低TTL阈值，以便流量能通过。

多条相等成本路径导致不需要的RPF行为

此部分显示组播源的等价路径如何能引起不需要的RPF行为。它也描述如何配置IP组播为了避免此行为。我们以下面的网络图为例。



诊断问题

在图，路由器75b有三等价路径回到来源(1.1.1.1)，并且选择您不希望是其第一选择作为RPF链路的链路。

当面对指向源的多个等价路径时，IP 多播会选择其独立多播协议 (PIM) 邻居具有最高 IP 地址的接口作为传入接口，然后将修剪发送给其他链路上的 PIM 邻居。

可能的修正

为了更改接口IP组播选择作为其流入接口，您能执行这些中的一个：

- 只在希望多播流经的接口上配置 PIM，这意味着要牺牲多播冗余。
- 更改子网，以使最高 IP 地址位于您希望作为多播主链路的链路上。
- 创建指出首选的组播接口，含义的组播路由(mroute)您失去组播冗余。

为举例说明，我们将创建一个静态多播路由。

在您安装一静态mroute前，您在此输出中看到路由表有源地址的1.1.1.1三等价路由。RPF 信息表明 RPF 接口为 E1/3：

```
ip22-75b#show ip route 1.1.1.1
Routing entry for 1.1.1.0/24
  Known via "ospf 1", distance 110, metric 20, type intra area
  Redistributing via ospf 1
  Last update from 3.1.1.1 on Ethernet1/2, 00:26:21 ago
  Routing Descriptor Blocks:
  * 4.1.1.1, from 10.0.119.66, 00:26:21 ago, via Ethernet1/3
    Route metric is 20, traffic share count is 1
  2.1.1.1, from 10.0.119.66, 00:26:21 ago, via Ethernet1/1
    Route metric is 20, traffic share count is 1
  3.1.1.1, from 10.0.119.66, 00:26:21 ago, via Ethernet1/2
    Route metric is 20, traffic share count is 1
```

```
ip22-75b#show ip rpf 1.1.1.1
RPF information for ? (1.1.1.1)
  RPF interface: Ethernet1/3
  RPF neighbor: ? (4.1.1.1)
  RPF route/mask: 1.1.1.0/24
  RPF type: unicast (ospf 1)
  RPF recursion count: 0
  Doing distance-preferred lookups across tables
```

```

ip22-75b#show ip mroute 224.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 01:30:34/00:02:59, RP 0.0.0.0, flags: DJC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/4, Forward/Sparse-Dense, 01:30:34/00:00:00
    Ethernet1/1, Forward/Sparse-Dense, 01:30:35/00:00:00
    Ethernet1/2, Forward/Sparse-Dense, 01:30:35/00:00:00
    Ethernet1/3, Forward/Sparse-Dense, 00:24:22/00:00:00

(1.1.1.1, 224.1.1.1), 01:30:35/00:02:59, flags: CT
  Incoming interface: Ethernet1/3, RPF nbr 4.1.1.1
  Outgoing interface list:
    Ethernet1/1, Prune/Sparse-Dense, 01:30:35/00:02:32
    Ethernet1/4, Forward/Sparse-Dense, 01:30:35/00:00:00
    Ethernet1/2, Prune/Sparse-Dense, 00:24:22/00:02:42

```

在您配置静态mroute后，您在输出RPF接口的这中看到更改对E1/1：

```

ip22-75b#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
ip22-75b(config)#ip mroute 0.0.0.0 0.0.0.0 2.1.1.1
ip22-75b(config)#end

ip22-75b#show ip rpf 1.1.1.1
RPF information for ? (1.1.1.1)
  RPF interface: Ethernet1/1
  RPF neighbor: ? (2.1.1.1)
  RPF route/mask: 1.1.1.1/0
  RPF type: static mroute
  RPF recursion count: 0
  Doing distance-preferred lookups across tables

ip22-75b#show ip route 1.1.1.1
Routing entry for 1.1.1.0/24
  Known via "ospf 1", distance 110, metric 20, type intra area
  Redistributing via ospf 1
  Last update from 3.1.1.1 on Ethernet1/2, 00:26:21 ago
  Routing Descriptor Blocks:
  * 4.1.1.1, from 10.0.119.66, 00:26:21 ago, via Ethernet1/3
    Route metric is 20, traffic share count is 1
  2.1.1.1, from 10.0.119.66, 00:26:21 ago, via Ethernet1/1
    Route metric is 20, traffic share count is 1
  3.1.1.1, from 10.0.119.66, 00:26:21 ago, via Ethernet1/2
    Route metric is 20, traffic share count is 1

```

```

ip22-75b#show ip mroute 224.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 01:31:29/00:02:59, RP 0.0.0.0, flags: DJC
  Incoming interface: Null, RPF nbr 0.0.0.0

```

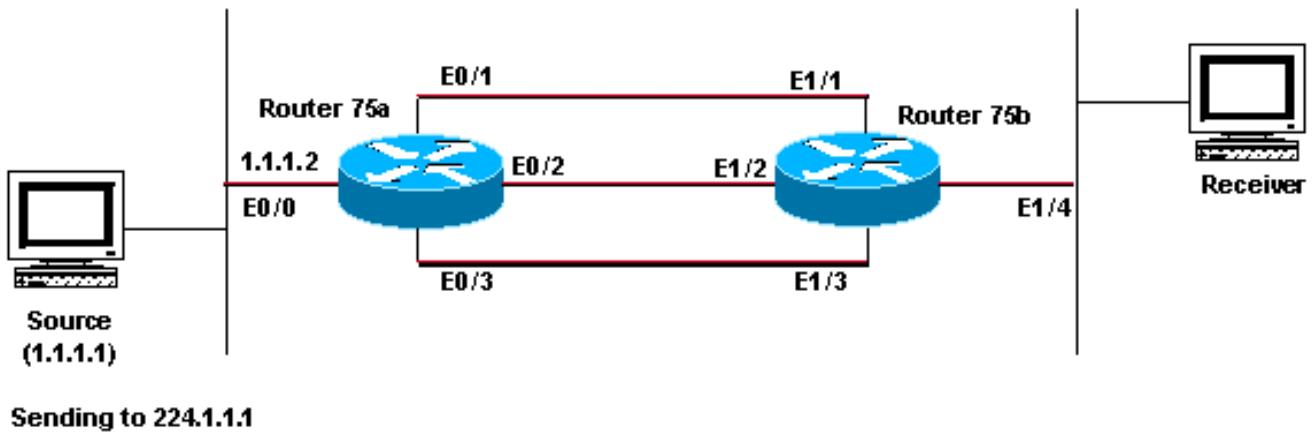
```
Outgoing interface list:
 Ethernet1/4, Forward/Sparse-Dense, 01:31:29/00:00:00
 Ethernet1/1, Forward/Sparse-Dense, 01:31:30/00:00:00
 Ethernet1/2, Forward/Sparse-Dense, 01:31:30/00:00:00
 Ethernet1/3, Forward/Sparse-Dense, 00:25:17/00:00:00
```

```
(1.1.1.1, 224.1.1.1), 01:31:30/00:02:59, flags: CT
 Incoming interface: Ethernet1/1, RPF nbr 2.1.1.1, Mroute
 Outgoing interface list:
 Ethernet1/4, Forward/Sparse-Dense, 01:31:30/00:00:00
 Ethernet1/2, Prune/Sparse-Dense, 00:25:17/00:01:47
 Ethernet1/3, Prune/Sparse-Dense, 00:00:31/00:02:28
```

为什么没有在所有可用的等价路径之间进行 IP 多播负载均衡？

此部分提供一解决方案给常见问题如何配置IP组播为了装载在所有可用的等价路径间的平衡。我们以下的网络图为例。

注意：在您装载在等价路径间的分开的IP组播流量在通道前，请配置CEF每个信息包负载均衡或者GRE数据包不会是每个小包被均衡的负载。关于对负载共享的其他方法在组播环境，请参阅[负载拆分在ECMP的IP组播流量](#)。



在图，路由器75b有三个等价路径回到来源(1.1.1.1)。您希望在全部三条链路上均衡多播流量。

可能的修正

如同您在[多条相等成本路径](#)看到了请[导致不需要的RPF行为](#)部分，独立于协议的组播(PIM)只选择RPF检查的一个接口并且修剪其他。这意味着不会进行负载均衡。要进行负载均衡，需要从冗余链路中删除 PIM 并在路由器 75a 与路由器 75b 之间创建一条隧道。然后您可以在链路级别进行负载均衡，并且 IP 将通过隧道运行。

下面是隧道的配置。

路由器 75a

```
interface Tunnel0
 ip address 6.1.1.1 255.255.255.0
 ip pim sparse-dense-mode
 tunnel source Ethernet0/0
 tunnel destination 5.1.1.1
!
interface Ethernet0/0
```

```

ip address 1.1.1.2 255.255.255.0
ip pim sparse-dense-mode
!
interface Ethernet0/1
ip address 2.1.1.1 255.255.255.0
!
interface Ethernet0/2
ip address 3.1.1.1 255.255.255.0
!
interface Ethernet0/3
ip address 4.1.1.1 255.255.255.0

```

路由器 75b

```

interface Tunnel0
ip address 6.1.1.2 255.255.255.0
ip pim sparse-dense-mode
tunnel source Ethernet1/4
tunnel destination 1.1.1.2
!
interface Ethernet1/1
ip address 2.1.1.2 255.255.255.0
!
interface Ethernet1/2
ip address 3.1.1.2 255.255.255.0
!
interface Ethernet1/3
ip address 4.1.1.2 255.255.255.0
!
interface Ethernet1/4
ip address 5.1.1.1 255.255.255.0
ip pim sparse-dense-mode
!
ip mroute 0.0.0.0 0.0.0.0 Tunnel0

```

在您配置通道后，请输入**show ip mroute**命令为了为组发现组播路由(mroute)：

```

ip22-75a#show ip mroute 224.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

(*, 224.1.1.1), 02:40:31/00:02:59, RP 0.0.0.0, flags: D
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:20:55/00:00:00

(1.1.1.1, 224.1.1.1), 02:40:32/00:03:29, flags: TA
  Incoming interface: Ethernet0/0, RPF nbr 0.0.0.0
  Outgoing interface list:
    Tunnel0, Forward/Sparse-Dense, 00:20:55/00:00:00

```

```

ip22-75b#show ip mroute 224.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode

```

```
(* , 224.1.1.1), 02:42:20/00:02:59, RP 0.0.0.0, flags: DJC
Incoming interface: Null, RPF nbr 0.0.0.0
Outgoing interface list:
Tunnel0, Forward/Sparse-Dense, 00:22:53/00:00:00
Ethernet1/4, Forward/Sparse-Dense, 02:42:20/00:00:00

(1.1.1.1, 224.1.1.1), 00:22:53/00:02:59, flags: CT
Incoming interface: Tunnel0, RPF nbr 6.1.1.1, Mroute
Outgoing interface list:
Ethernet1/4, Forward/Sparse-Dense, 00:22:53/00:00:00
```

为了检查组播数据是负载在三条链路间均等地平衡了，查看路由器75a接口数据。

传入接口为 9000 位/秒：

```
ip22-75a#show interface e0/0
.
.
5 minute input rate 9000 bits/sec, 20 packets/sec
```

三个传出接口每个均显示 3000 位/秒：

```
ip22-75a#show interface e0/1
.
.
5 minute output rate 3000 bits/sec, 7 packets/sec
```

```
ip22-75a#show interface e0/2
.
.
5 minute output rate 3000 bits/sec, 7 packets/sec
```

```
ip22-75a#show interface e0/3
.
.
5 minute output rate 3000 bits/sec, 7 packets/sec
```

为什么在路由器上收到 IP 多播“INVALID_RP_JOIN”错误消息？

此部分为常见的 IP 多播“INVALID_RP_JOIN”错误消息提供了一个解决方案。

诊断问题 - 第 1 部分

此错误消息在聚合点(RP)接收：

```
00:55:15: %PIM-6-INVALID_RP_JOIN: Received (*, 224.1.1.1)
Join from 1.1.6.2 for invalid RP 1.1.5.4
00:56:15: %PIM-6-INVALID_RP_JOIN: Received (*, 224.1.1.1)
Join from 1.1.6.2 for invalid RP 1.1.5.4
```

[Cisco IOS Software System Error Messages](#) (Cisco IOS 软件系统错误消息) 解释了此错误的原因：一个下游 PIM 路由器发出加入共享树的消息，但此路由器并不希望接受。这一行为表明此路由器只允许下游路由器加入特定 RP。

这里可能存在某种过滤。您需要查看一下路由器的配置：

```
interface Ethernet0/0
ip address 1.1.5.4 255.255.255.0
ip pim sparse-dense-mode
```

```
!  
ip pim accept-rp 10.2.2.2 8  
ip pim send-rp-announce Ethernet0/0 scope 15  
ip pim send-rp-discovery scope 15  
!  
access-list 8 permit 224.0.0.0 0.255.255.255
```

那么配置中的 **accept-rp** 语句的用途是什么呢？在 [IP Multicast Routing Commands](#) (IP 多播路由命令) 中，该文档说明“要配置路由器以接受指定 RP 和特定组列表的‘加入’或‘修剪’设置，可以使用 **ip pim accept-rp** 全局配置命令。要删除这项检查，可使用此命令的 **no** 格式。”

如果删除 **ip pim accept-rp** 命令，就会出现错误消息。引起问题若找到命令，但是您要保持该in命令配置？您也许允许错误的RP地址。输入**show ip pim rp mapping**命令为了发现正确RP地址：

```
ip22-75a#show ip pim rp mapping  
PIM Group-to-RP Mappings  
This system is an RP (Auto-RP)  
This system is an RP-mapping agent  
  
Group(s) 224.0.0.0/4  
  RP 1.1.5.4 (?), v2v1  
    Info source: 1.1.5.4 (?), via Auto-RP  
    Uptime: 01:00:48, expires: 00:02:07
```

根据输出，1.1.5.4是通过自动RP了解的唯一的RP或。然而，此路由器是组 224.0.0.0/4 的唯一RP。因此，**pim accept-rp**语句在配置里是错误的。

可能的修正

解决方案是纠正 **ip pim accept-rp** 语句中的 IP 地址，如下所示：

将此语句：

```
ip pim accept-rp 10.2.2.2 8  
对此：
```

```
ip pim accept-rp 1.1.5.4 8
```

您能也更改语句接受什么在Auto-RP缓存和确定访问列表(8在本例中)允许必要的组播组范围。示例如下：

```
ip pim accept-rp auto-rp  
  
access-list 8 permit 224.0.0.0 0.255.255.255
```

诊断问题 - 第 2 部分

此错误消息出现在 router2 上。

```
router2#  
*Aug 12 15:18:17.891:  
  %PIM-6-INVALID_RP_JOIN: Received (*, 224.0.1.40)  
  Join from 0.0.0.0 for invalid RP 2.1.1.1
```

检查以确认 router2 是否是组 224.1.1.1 的 RP：

```
router2#show ip pim rp map  
PIM Group-to-RP Mappings  
  
Group(s) 224.0.0.0/4  
  RP 1.1.1.1 (?), v2v1
```



```
Info source: 1.1.1.1 (?), elected via Auto-RP
Uptime: 00:21:26, expires: 00:02:24
```

```
router2#
```

224.1.1.1 的 RP 是 1.1.1.1。

检查它是否是 router2 的其中一个接口：

```
router2#show ip interface brief
```

Interface	IP-Address	OK?	Method	Status	Protocol
Ethernet0/0	1.1.1.2	YES	NVRAM	up	up
Ethernet1/0	2.1.1.1	YES	NVRAM	up	up
Ethernet2/0	unassigned	YES	NVRAM	administratively down	down

```
router2#
```

由于 router2 不是 RP，因此不应收到此 RP-Join 数据包。检查下游路由器为什么将“加入”发送给 router2，本来不应如此：

```
router3#show ip pim rp map
```

```
PIM Group-to-RP Mappings
```

```
Group(s) 224.0.0.0/4
```

```
RP 1.1.1.1 (?), v2v1
```

```
Info source: 1.1.1.1 (?), elected via Auto-RP
```

```
Uptime: 00:24:30, expires: 00:02:16
```

```
Group(s): 224.0.0.0/4, Static-Override
```

```
RP: 2.1.1.1 (?)
```

```
router3#
```

您看到，router3 静态配置 RP 信息和点对 router2，不正确。这解释 router3 为什么发送 RP-Join 对 router2。

可能的修正

将 router2 作为组 224.1.1.1 的 RP，或者更改 router3 的设置使其引用正确的 RP 地址。

```
router3#show run | i rp
```

```
ip pim rp-address 2.1.1.1 override
```

```
router3#configure terminal
```

```
Enter configuration commands, one per line. End with CNTL/Z.
```

```
router3(config)#no ip pim rp-address 2.1.1.1 override
```

```
router3(config)#exit
```

```
router3#
```

更正 router3 的配置后，router3 将引用正确的 RP 地址，这时便会停止显示错误消息。

```
router3#show ip pim rp map
```

```
PIM Group-to-RP Mappings
```

```
Group(s) 224.0.0.0/4
```

```
RP 1.1.1.1 (?), v2v1
```

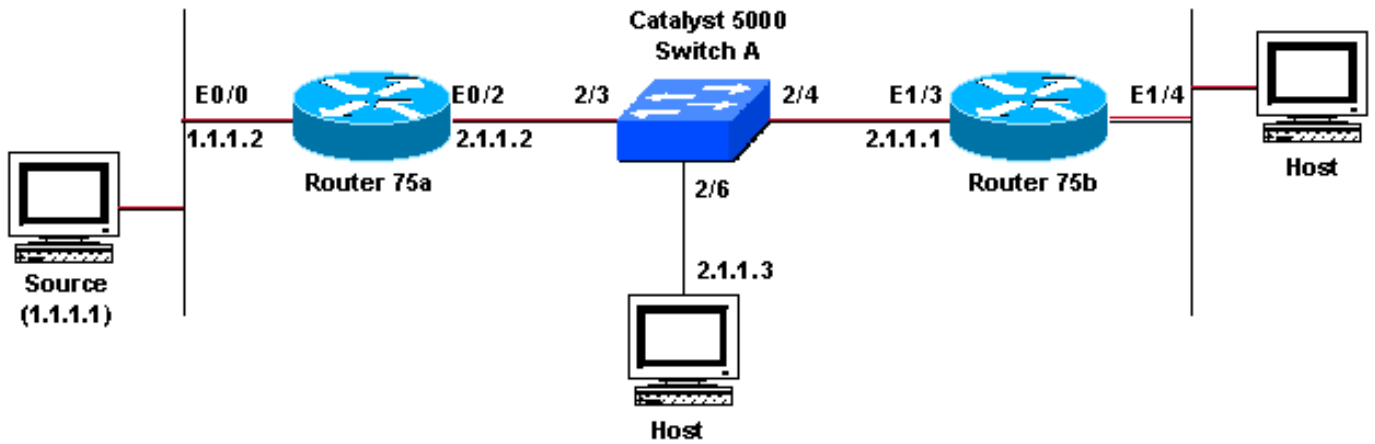
```
Info source: 1.1.1.1 (?), elected via Auto-RP
```

```
Uptime: 00:30:45, expires: 00:02:02
```

```
router3#
```

CGMP 不能防止多播数据包泛洪

此部分说明组播信息包不需要的泛滥如何能发生，当思科组管理协议(CGMP)在特定子网的时所有路由器没有启用，并且此问题如何可以避免。我们以下面的网络图为例。



诊断问题

在图，在Catalyst 5000交换机A的静态CAM表不显示填充的其中一个正确端口。配置了 CGMP 的路由器未发送 CGMP 数据包。

Switch A 上的 **set cgmp enable** 命令和路由器 75a 的 E0/2 接口上的 **ip cgmp** 命令已经正确配置了 CGMP。然而，当发出 **show multicast group** 命令时，在 Switch A 上看不到多播组。

```
Console> (enable) show multicast group
CGMP enabled
IGMP disabled
IGMP fastleave disabled
GMRP disabled
```

```
VLAN  Dest MAC/Route Des      [CoS]  Destination Ports or VCs / [Protocol Type]
-----
```

Total Number of Entries = 0

此命令的输出应当显示其中具有配置了 CGMP 的路由器的每个端口 (端口 2/3) 以及其中具有意向接收方的每个端口 (端口 2/6)。由于列出了 0，因此不管希望与否，多播流量都可能会不必要地泛洪到所有端口。

观察

检查路由器 75a 上是否有任何独立多播协议 (PIM) 邻居：

```
ip22-75a#show ip pim neighbor
PIM Neighbor Table
Neighbor Address  Interface          Uptime    Expires    Ver  Mode
2.1.1.1           Ethernet0/2        00:07:41  00:01:34  v2
```

输出显示路由器75a能发现路由器75b作为一个有效PIM邻居通过交换机A。

现在请证实您是否获得关于路由器的正确组播路由(mroute)信息：

```
ip22-75a#show ip mroute 224.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(* , 224.1.1.1), 00:14:55/00:02:59, RP 0.0.0.0, flags: DJC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/2, Forward/Sparse-Dense, 00:14:55/00:00:00
```

```
(1.1.1.1, 224.1.1.1), 00:14:56/00:02:59, flags: CTA
  Incoming interface: Ethernet0/1, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet0/2, Forward/Sparse-Dense, 00:14:56/00:00:00
```

输出显示路由器75a转发组播信息包E0/2往交换机A。此输出显示路由器75b得到组播信息包并且正确地转发他们：

```
ip22-75b#show ip mroute 224.1.1.1
IP Multicast Routing Table
Flags: D - Dense, S - Sparse, C - Connected, L - Local, P - Pruned
       R - RP-bit set, F - Register flag, T - SPT-bit set, J - Join SPT
       M - MSDP created entry, X - Proxy Join Timer Running
       A - Advertised via MSDP
Timers: Uptime/Expires
Interface state: Interface, Next-Hop or VCD, State/Mode
```

```
(* , 224.1.1.1), 00:17:57/00:02:59, RP 0.0.0.0, flags: DJC
  Incoming interface: Null, RPF nbr 0.0.0.0
  Outgoing interface list:
    Ethernet1/3, Forward/Sparse-Dense, 00:17:57/00:00:00
    Ethernet1/4, Forward/Sparse-Dense, 00:17:57/00:00:00
```

```
(1.1.1.1, 224.1.1.1), 00:00:35/00:02:59, flags: CTA
  Incoming interface: Ethernet1/3, RPF nbr 2.1.1.2
  Outgoing interface list:
    Ethernet1/4, Forward/Sparse-Dense, 00:00:35/00:00:00
```

从交换机A的观点，您看到看到路由器75a端口2/3。

```
Console> (enable) show multicast router
CGMP enabled
IGMP disabled
IGMP fastleave disabled
```

Port	Vlan
2/3	6

Total Number of Entries = 1

到目前为止一切正常。输入一些调试on命令路由器75a为了发现什么您能发现：

```
ip22-75a#debug ip cgmp
CGMP debugging is on
*Feb  8 12:45:22.206: CGMP: Sending self Join on Ethernet0/2
*Feb  8 12:45:22.206:      GDA 0000.0000.0000, USA 00d0.ff2f.a002
*Feb  8 12:46:22.234: CGMP: Sending self Join on Ethernet0/2
*Feb  8 12:46:22.234:      GDA 0000.0000.0000, USA 00d0.ff2f.a002
*Feb  8 12:47:22.262: CGMP: Sending self Join on Ethernet0/2
*Feb  8 12:47:22.262:      GDA 0000.0000.0000, USA 00d0.ff2f.a002
```

在输出中，当路由器发送CGMP加入/离开消息，因此交换机能填充路由器端口时，0000.0000.0000是所有组地址和使用。组目的地址的GDA立场在媒体访问控制(MAC)级别格式和单播源地址的USA立场。这是产生IGMP报告此CGMP信息生成主机的地址。在本例中，这是路由器75a接口E0/2的MAC地址。可以使用show interface命令查看路由器75a的E0/2的MAC地址，如下所示：

```
ip22-75a#show interface e0/2
```

```
Ethernet0/2 is up, line protocol is up
  Hardware is cxBus Ethernet, address is 00d0.ff2f.a002 (bia 00d0.ff2f.a002)
```

另外，当debug ip igmp命令打开时，您也许周期地发现此：

```
*Feb  8 12:51:41.854: IGMP: Received v2 Report from 2.1.1.3 (Ethernet0/2) for 224.1.1.1
```

然而，随后不会看到来自路由器 75a 的对应 CGMP 数据包。这意味着路由器75a接收IGMP报告，但是不生成必要的CGMP信息包来帮助交换机A知道阻塞的哪些端口。如果路由器 75a 是 IGMP 查询方，则这是所期望的。从路由器75a的此输出告诉我们预料之中的行为为什么不出现：

```
ip22-75a#show ip igmp interface e0/2
Ethernet0/2 is up, line protocol is up
  Internet address is 2.1.1.2/24
  IGMP is enabled on interface
  Current IGMP version is 2
  CGMP is enabled
  IGMP query interval is 60 seconds
  IGMP querier timeout is 120 seconds
  IGMP max query response time is 10 seconds
  Last member query response interval is 1000 ms
  Inbound IGMP access group is not set
  IGMP activity: 3 joins, 1 leaves
  Multicast routing is enabled on interface
  Multicast TTL threshold is 0
  Multicast designated router (DR) is 2.1.1.2 (this system)
  IGMP querying router is 2.1.1.1
  No multicast groups joined
```

如果同一子网中有两个路由器，并且您将它们都配置为 CGMP，则只有一个路由器会发送 CGMP 数据包。发送 CGMP 数据包的路由器是 IGMP 查询路由器。这意味着该路由器是启用了 IGMP 的路由器中具有最低单播 IP 地址的路由器。

在本例中，路由器 75a 和路由器 75b 都启用了 IGMP (路由器 75b 成为 IGMP 查询路由器)，但是只有路由器 75a 启用了 CGMP。因为路由器75a不是IGMP查询路由器，CGMP信息包没有被发送。

可能的修正

为了解决问题，您需要配置在IGMP查询路由器的CGMP。本例中是路由器 75b。首先，对路由器 75b 执行 debug 命令：

```
ip22-75b#conf t
Enter configuration commands, one per line.  End with CNTL/Z.
ip22-75b(config)#int e 1/3
ip22-75b(config-if)#ip cgmp
ip22-75b(config-if)#end
ip22-75b#debug ip cgmp
ip22-75b#debug ip igmp
*Feb  8 12:58:42.422: IGMP: Received v2 Report from 2.1.1.3 (Ethernet1/3) for 224.1.1.1
*Feb  8 12:58:42.422: CGMP: Received IGMP Report on Ethernet1/3
*Feb  8 12:58:42.422:      from 2.1.1.3 for 224.1.1.1
*Feb  8 12:58:42.422: CGMP: Sending Join on Ethernet1/3
*Feb  8 12:58:42.422:      GDA 0100.5e01.0101, USA 0030.b655.a859
```

路由器 75b 从 2.1.1.3 收到组 224.1.1.1 的 IGMP 报告。随后，它为 224.1.1.1 的等价 MAC 连同 2.1.1.3 意向主机的 MAC 地址 (USA) 向交换机 A 发送一个 CGMP“加入”。交换机 A 知道主机位于哪个端口，因此会将其标记为打开并禁用其他端口。

现在，交换机 A 上应当一切正常了：

```
Console> (enable) show multicast group
```

```

CGMP enabled
IGMP disabled
IGMP fastleave disabled
GMRP disabled

```

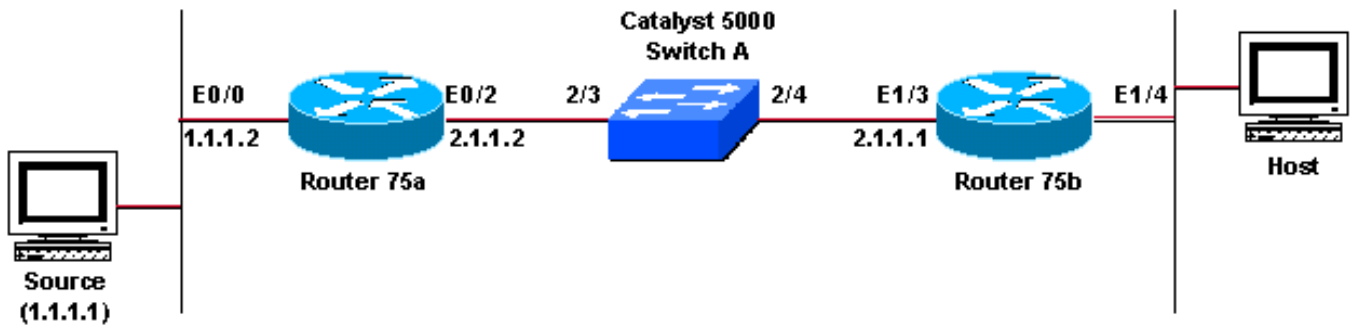
VLAN	Dest MAC/Route Des	[CoS]	Destination Ports or VCs / [Protocol Type]
6	01-00-5e-00-01-28		2/3-4
6	01-00-5e-01-01-01		2/3-4,2/6

Total Number of Entries = 2

现在情况好多了。224.1.1.1 (01-00-5e-01-01-01)数据包是交换机A的仅被派出的端口2/3，2/4和2/6，他们应该。向所有其他端口的泛洪已经停止。现在所列的条目总数也是正确的（为2）。MAC地址01-00-5e-00-01-28映射自多播地址224.0.1.40，用于CGMP“自我加入”。

由于源端/接收端的放置问题，CGMP不能防止多播数据包泛洪

此部分提供一解决方案给对每个端口的溢出数据流，正确主机的Catalyst交换机的常见问题。我们以下面的网络图为例。



诊断问题

在图、路由器75a和75b和Catalyst 5000 (交换机A)为组播和思科组管理协议(CGMP)适当地配置。主机获得组播数据流。然而，如此是其他主机交换机A.交换机A充斥流量每个端口，因此意味着CGMP不工作。

交换机 A 的 `show multicast group` 命令的输出为：

```

Console> (enable) show multicast group
CGMP enabled
IGMP disabled
IGMP fastleave disabled
GMRP disabled

```

VLAN	Dest MAC/Route Des	[CoS]	Destination Ports or VCs / [Protocol Type]
6	01-00-5e-00-01-28		2/3-4

Total Number of Entries = 1

您能从输出看到唯一的组是224.0.1.40，路由器使用，当发送自动RP组的时CGMP自加入。为什么没有其他组？

可能的修正

为了了解解决方案，您需要知道CGMP在一定条件下如何正常运行。支持CGMP的路由器发送CGMP加入到交换机通知在特定组播组中感兴趣的交换机主机。交换机在其CAM表中查找这些主机的MAC地址，然后将多播数据包从意向主机的端口转发出去并禁止所有其他端口转发多播数据包。

如果路由器所接收的多播数据包的源位于启用了CGMP的相同接口上，则它会从启用了CGMP的接口发出CGMP“自我加入”。

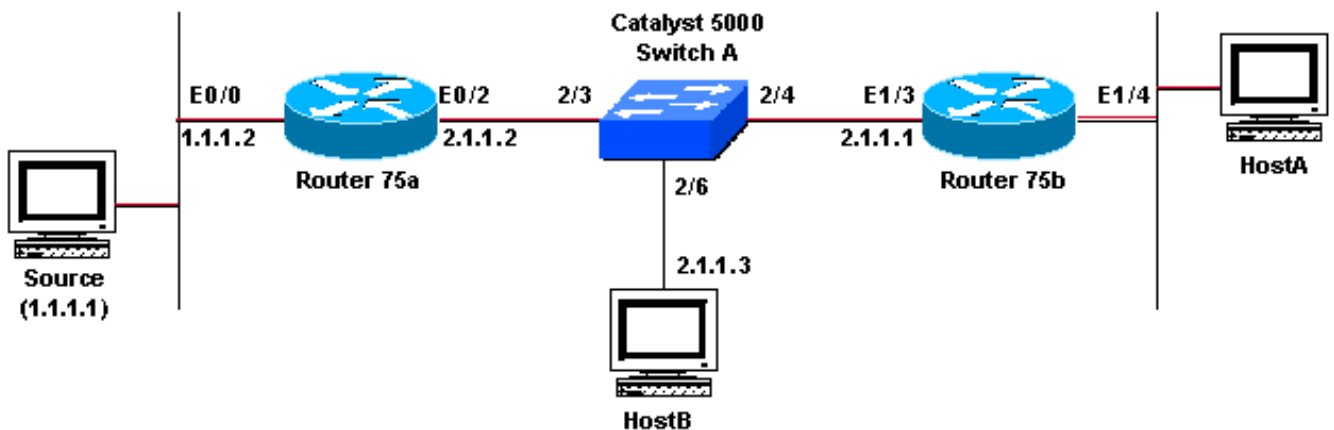
例如，如果源位于与路由器75a和75b相同的子网(VLAN)2.1.1.0/24上，则CGMP将正常工作。路由器在看到来自源的数据包时，将向交换机发送一个CGMP“自我加入”，然后交换机将动态确定路由器所在的端口并禁止所有其他端口转发多播数据包。

如果路由器所接收的IGMP报告的源主机位于启用了CGMP的相同接口上，则它会从启用了CGMP的接口发出CGMP“加入”。

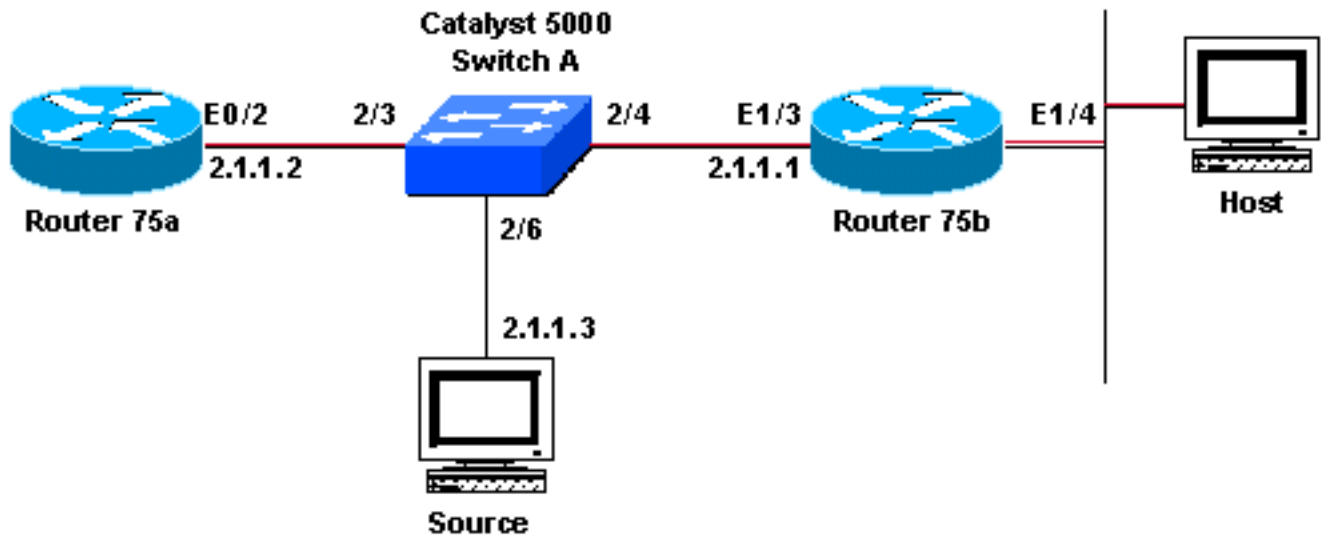
如果有在此同样VLAN的一响应主机另一示例是。在那种情况下，当支持CGMP的路由器接收从是在特定组播组中感兴趣的主机的一互联网组管理协议(IGMP)报告，路由器将发送CGMP加入。加入将指示要加入要参加主机和组的MAC控制(MAC)地址。然后，Catalyst 5000将在其CAM表中检查该主机的MAC地址，将关联端口放到多播组列表中，并禁止所有其他无意的端口。

当源和意向主机位于启用了CGMP的子网(VLAN)以外的其他子网上时，情况就不是这样了。组播信息包，那自来源来，不触发路由器发送CGMP自加入到交换机。因此，数据包到达交换机后将在VLAN中泛洪到各处。此方案继续直到在VLAN的一台主机，那结束在交换机的端口，发送IGMP加入。只有在收到IGMP报告后，路由器才会发送CGMP数据包，进而导致交换机添加适当主机端口作为转发端口，并禁用(除了路由器端口以外的)所有其他端口。

因此，要使CGMP在此中转类型拓扑中正常工作，可以向路由器75a和75b所在的相同VLAN中添加一个主机，如下面的网络图所示。



也可以将源移到路由器75a和75b所在的相同子网上，如下面的示例所示。



将源移到相同的子网中，然后检查交换机 A 的输出：

```
Console> (enable) show multicast router
CGMP enabled
IGMP disabled
IGMP fastleave disabled
```

```
Port      Vlan
-----  -----
2/3      6
2/4      6
```

```
Total Number of Entries = 2
'*' - Configured
Console> (enable)
```

```
Console> (enable) show multicast group
CGMP enabled
IGMP disabled
IGMP fastleave disabled
GMRP disabled
```

```
VLAN  Dest MAC/Route Des  [CoS]  Destination Ports or VCs / [Protocol Type]
-----  -----  -----  -----
6      01-00-5e-00-01-28  2/3-4
6      01-00-5e-01-01-01  2/3-4
```

```
Total Number of Entries = 2
```

现在 224.1.1.1 (01-00-5e-01-01-01) 只会泛洪到路由器端口 2/3 和 2/4，而不是交换机 A 的所有传出端口。

CGMP 不能防止某些组地址的多播数据包泛洪

此部分说明一些组播IP地址为什么造成思科组管理协议(CGMP)充斥组播数据流局域网的所有端口。当您使用组播组地址225.0.0.1时，CGMP不工作。它会将多播流从所有交换机端口泛出并浪费带宽。然而，如果将地址更改为 225.1.1.1，CGMP 将会正常发挥作用。225.0.0.1 并不是路由协议的注册地址，那么使用它有什么错误呢？

可能的修正

首先，我们必须清楚第 3 层多播地址是如何映射到第 2 层多播地址的。所有 IP 组播帧使用开始与 0x0100.5e 24 位前缀的 IEEE MAC 控制层地址。当您映射第 3 层对 Layer2 地址时，第 3 层组播地址的低价位 23 个位被映射到 IEEE MAC 地址的低价位 23 个位。

要了解的另一个重要事实是那里是 28 个位 IP 组播地址的(减包含 1110 D 类前缀)的前 4 个位的 32 个位唯一地址空间。由于只有 23 位插入到 IEEE MAC 地址中，因此会保留 5 位重叠部分。这意味着多个第 3 层多播地址会映射到相同的第 2 层多播地址。

例如：

```
224.0.0.1 = 1110 0000.0000 0000.0000 0000.0000 0001 in binary
low order 23 bits =    000 0000.0000 0000.0000 0001
hex equivalent    =    0   0   0   0   0   1
result of mapping = 0x0100.5e00.0001
```

```
224.128.0.1 = 1110 0000.1000 0000.0000 0000.0000 0001 in binary
low order 23 bits =    000 0000.0000 0000.0000 0001
hex equivalent    =    0   0   0   0   0   1
result of mapping = 0x0100.5e00.0001
```

注意在示例 224.0.0.1 和 224.128.0.1 地图对同一个 Layer2 组播地址。

既然您知道对 Layer2 组播地址的第 3 层如何被映射，请继续对特定解决方案对此问题。

交换机 A 充斥组播信息包对 224.0.0.x，因为那些地址是链路本地，并且您要确保链路本地地址获得对在局域网的所有设备。Catalyst 交换机还会将多播数据包泛洪到 MAC 级别的其他 224.0.0.x 通配多播地址（例如 224.0.0.1 和 225.0.0.1 都会映射到 0100.5e00.0001）。交换机会将多播数据包泛洪到这些本地链路地址而不管是否启用了 CGMP。

因此，多播应用必须避免使用映射到第 2 层多播地址 0100.5e00.00xx 的 D 类地址，其中 xx 可以是十六进制的 00 到 FF。这包括下列 D 类地址：

```
224.0.0.x (x = 0 to 255)
225.0.0.x
.
239.0.0.x
```

```
224.128.0.x (x = 0 to 255)
225.128.0.x
.
239.128.0.x
```

收到重复的多播数据包流

原因 1

当在密集模式下配置了两个路由器时就会收到重复的多播数据包。在密集模式下，设备会定期泛洪流。泛洪后，它会修剪掉不需要流的接口。两个路由器也将经历该确认过程并确定哪一个是转发者。在计时器去此发生时候，并且直到此进程完成，两路由器转发数据流。这会致应用程序收到重复的多播流。

可能的修复方法 1

如果将其中一个路由器配置为多播路由，另一个路由器配置为上游 RP，则可以解决此问题。将其

配置为在流进入路由器之前将流转换为稀疏模式。这可以防止重复的数据包抵达应用程序。网络并不负责确保不会有重复的数据包抵达终端主机。应用程序将负责处理重复的数据包并忽略不需要的数据。

原因 2

该问题会在 Cisco Catalyst 6500 交换机中出现，这些交换机配置为出口多播复制模式，可通过移除并重新插入任何板卡 [OIR] 进行触发。在 OIR 以后，退出 [FPOE] 结构端口可以是编程错误，能造成数据包被导向到错误的结构退出信道和被发送到错误的线路卡。结果是数据包沿环路返回光纤通道，并且当其在正确的板卡上退出时将重复多次。

```
C6509#show mls ip multicast capability
Current mode of replication is Egress
Auto replication mode detection is ON
```

Slot	Multicast replication capability
1	Egress
2	Egress
3	Egress
4	Egress
5	Egress
6	Egress
7	Egress

可能的修复方法 2

要解决此问题，可以更改为入口复制模式。在从出口复制模式更改为入口复制模式的过程中会出现流量中断，因为要清除并重新安装快捷方式。

```
mls ip multicast replication-mode ingress
```

请将 Cisco IOS 软件升级到不受 Cisco Bug ID [CSCeg28814](#) ([仅限注册客户](#)) 影响的版本以永久解决此问题。

原因 3

此问题能也出现，如果接收端比例缩放(RSS)设置，在终端-主机或服务器，禁用。

可能的修复方法 3

RSS 设置有利于在多个 CPU 间快速传输数据。请在终端主机或服务器上启用 RSS 设置。参考[与 RSS 的 Microsoft 条款可扩展网络](#) 欲知更多信息。

多播数据包为什么被丢弃？

原因 1

在传输多播流量时，可能会发现流量过大并且输入数据包丢弃在接口上的情况。您可以使用 **show interface** 命令检查流量。

```
Switch#show interface gi 1/0
```

```
!--- Output suppressed MTU 1500 bytes, BW 1000000 Kbit, DLY 10 usec, reliability 255/255, txload 1/255, rxload 1/255 Encapsulation ARPA, loopback not set Keepalive set (10 sec) Full-duplex,
```

1000Mb/s, media type is SX input flow-control is off, output flow-control is on Clock mode is auto ARP type: ARPA, ARP Timeout 04:00:00 Last input 00:00:00, output 00:00:00, output hang never Last clearing of "show interface" counters never [Input queue: 2/75/0/13370328 \(size/max/drops/flushes\)](#); Total output drops: 0 Queueing strategy: fifo Output queue: 0/40 (size/max) 30 second input rate 195000 bits/sec, 85 packets/sec 30 second output rate 1000 bits/sec, 1 packets/sec L2 Switched: ucast: 53056 pkt, 4728434 bytes - mcast: 235759386 pkt, 66914376970 bytes L3 in Switched: ucast: 8714263 pkt, 1815426423 bytes - mcast: 1081138213 pkt, 438000092206 bytes mcast L3 out Switched: ucast: 4939256 pkt, 790351689 bytes mcast: 0 pkt, 0 bytes 1326976857 packets input, 506833655045 bytes, 0 no buffer Received 1318209538 broadcasts, 0 runts, 0 giants, 1 throttles 0 input errors, 0 CRC, 0 frame, 0 overrun, 0 ignored 0 input packets with dribble condition detected 31643944 packets output, 3124494549 bytes, 0 underruns 0 output errors, 0 collisions, 1 interface resets 0 babbles, 0 late collision, 0 deferred 0 lost carrier, 0 no carrier 0 output buffer failures, 0 output buffers swapped out

可能的修复方法 1

对于发现过大流量的接口，可以将 SPT 值设置为无限。

进行如下配置：

```
Switch(config-if)#ip pim spt-threshold infinity
```

原因 2

当您使用 `ip igmp join-group <group name> on` 命令所有接口时，处理交换。如果在任何接口上对多播数据包进行进程交换，则会消耗更多 CPU，因为这会强制要求该组的所有数据包都进行进程交换。您可以运行 `show buffers input-interface` 命令并检查异常大小。

```
Switch#show buffers input-interface gi 1/0
```

Header	DataArea	Pool	Rcnt	Size	Link	Enc	Flags	Input	Output
437C6EAC	8096AE4	Middl	1	434	7	1	280	Gi1/1	None
437C74B4	8097864	Middl	1	298	7	1	280	Gi1/1	None
437C98E4	809C964	Middl	1	434	7	1	280	Gi1/1	None
437CA AFC	809F1E4	Middl	1	349	7	1	280	Gi1/1	None
437CAE00	809F8A4	Middl	1	519	7	1	280	Gi1/1	None

!--- Output suppressed

可能的修复方法 2

您可以使用 `ip igmp static-group <group-name>` 命令而不是 `ip igmp join-group <group-name>` 命令。

注意：由于前面提到的问题，很可能会看到 CPU 的使用率高达 90% 左右。通过可能的修复方法解决这些问题后，CPU 使用率将下降到正常水平。

相关信息

- [基本多播故障排除工具](#)
- [组播快速开始配置指南](#)
- [IP 多播技术支持](#)
- [IP 路由协议支持](#)
- [技术支持&说明文件Cisco系统](#)